UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CURSO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMAÇÃO DE COMPUTADORES

LABORATÓRIO 4

MODULARIDADE E FUNÇÕES

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. As funções `rand()` e `srand()` da biblioteca `<cstdlib>` funcionam em conjunto para gerar números pseudoaleatórios. Enquanto *rand* gera um número pseudoaleatório entre 0 e 32767, *srand* gera uma nova semente para *rand*. A semente é o valor inicial usado pelo gerador para criar os números. Para uma dada semente, a sequência de números gerados por *rand* é sempre a mesma.

   Para entender melhor o funcionamento dessas funções, crie um programa que use `srand(1)`, e em seguida exiba o resultado a cinco chamadas da função `rand()`. Execute o programa várias vezes. O resultado é sempre o mesmo?

   ```
   Gerando números pseudoaleatórios:
   41 18467 6334 26500 19169
   ```

   Agora mude o valor passado a `srand()` e tente executar novamente o programa. O resultado obtido foi diferente? Execute o programa várias vezes.

2. Uma forma de obter um número pseudoaleatório mais imprevisível é usar um valor de semente que mude com o tempo. A biblioteca `<ctime>` possui uma função `time()` que retorna uma quantidade de segundos transcorrida (geralmente desde 1 de janeiro de 1970) quando passamos `NULL` como argumento da função.

   ```
   srand(time(NULL));
   ```

   Teste o resultado dessa modificação criando um número pseudoaleatório com `rand()` e rodando o programa várias vezes.

   Por fim, exiba a mensagem GRANDE se o número obtido for maior que 16834 e PEQUENO caso contrário.

   ```
   Gerando número pseudoaleatório:
   3198
   PEQUENO
   ```

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Escreva um programa que produza a saída abaixo. Para atingir esse objetivo crie uma função que exiba "Sorria!" uma única vez e chame-a na função principal quantas vezes for preciso para gerar a saída.

```
Sorria! Sorria! Sorria! Sorria!
Sorria! Sorria!
Sorria!
```

   Dica: o salto de linha não pode estar dentro da função.

2. Escreva uma função `Linha()` que, usando um único cout e sem usar repetições ou saltos de linha, produza uma linha de tamanho fixo formada por 10 caracteres de hífen. Escreva também uma função `Pequena()`, `Media()` e `Grande()` que façam uso da função `Linha()` e de saltos de linha para obter linhas com 10, 20 e 30 hifens respectivamente. O programa deve produzir a saída abaixo. A função `Linha()` não deve ser chamada diretamente pela função `main()`, apenas `Pequena()`, `Media()` e `Grande()` devem fazer uso dela.

```
----------
--------------------
------------------------------
Programação de Computadores
------------------------------
--------------------
----------
```

3. Escreva um programa que chame uma função de nome `UmTres()`. Esta função deve exibir na tela a palavra "Um", chamar a função de nome `Dois()`, e então exibir a palavra "Três". A função `Dois()` deve exibir a palavra "Dois" na tela. A função `main()` deve mostrar a frase "Começando agora:" , chamar a função `UmTres()` e em seguida mostrar a palavra "Pronto!".

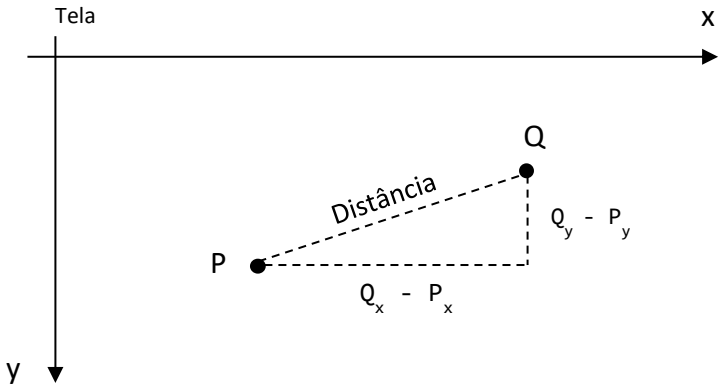
   A saída deve ser a seguinte:

```
Começando agora:
Um Dois Três
Pronto!
```

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Dados dois pontos P e Q na tela do computador, calcule a distância entre eles utilizando a fórmula: $Distância = \sqrt{(Qx - Px)^2 + (Qy - Py)^2}$




```
Ponto P:
5 8
Ponto Q:
20 4

A distância entre P e Q é: 15.5242
```

Dica: utilize as funções sqrt e pow da biblioteca cmath.

2. Construa um programa que determine o seno de um ângulo dado pelo usuário. Para achar o seno utilize a função sin(), definida na biblioteca cmath. Pesquise a função sin() na Internet para saber como utilizá-la, isto é, para saber os tipos dos parâmetros e valor de retorno.

```
Digite um ângulo: 90
Seno = 1
```

3. Escreva uma função chamada media() que recebe dois valores inteiros e retorna a média aritmética destes valores. Teste a função com um programa simples.

```
Digite um valor inteiro: 10
Digite outro valor inteiro: 11
A média dos números é 10.5
```

4. O volume de um cilindro é dado pela fórmula $V = \pi r^2 h$, onde r é o raio da base, h é a altura do cilindro e π é 3.14159. Escreva uma função chamada VolumeCilindro() que recebe a altura e o raio da base e retorna o volume do cilindro. Escreva um programa principal que leia a altura e o raio de um cilindro, use a função VolumeCilindro() para achar o seu volume e então mostre o resultado na tela.

```
Calcula o Volume de um Cilindro
-------------------------------
Entre com o raio da base: 10
Entre com a altura: 5
O volume do cilindro é 1570.795
```

5. Escreva um programa que peça ao usuário para digitar um número inteiro (positivo ou negativo). Crie e use uma função chamada Absoluto() para encontrar o valor absoluto de um número. A função Absoluto() deve receber um valor inteiro por parâmetro e retornar o seu valor absoluto.

```
Digite um número inteiro: -10
O valor absoluto é 10.
```

Dica: uma forma de achar o valor absoluto de um número é elevando-o ao quadrado e depois achando a sua raiz quadrada. A função Absoluto() deve implementar esta técnica usando as funções pow() e sqrt() da biblioteca cmath.

6. Escreva uma função Inicializar() que **se utiliza de quatro outras funções** para cumprir sua tarefa: a função deve escrever a mensagem "Inicializando sistema:", chamar a função ligar(), verificar() e ativar() e em seguida escrever "Inicialização concluída". Para simular o resultado da inicialização, a função Inicializar() deve retornar um valor inteiro aleatório, obtido com uma chamada da função rand().

   A função rand() pertence a biblioteca <cstdlib> e retorna um número aleatório entre 0 e 32767 cada vez que é chamada. As funções ligar(), verificar() e ativar() devem ser criadas para exibir mensagens, como no exemplo abaixo.

   Caso o resultado retornado por Inicializar() seja maior que 16384, o programa deve exibir "Sistema em funcionamento", caso contrário ele deve exibir "Falha na inicialização".

```
Inicializando Sistema:
- Ligando dispositivos
- Verificando integridade
- Ativando processos
Inicialização concluída.

Sistema em funcionamento.
```